COMISIÓN NACIONAL DE BANCOS Y SEGUROS

WORKING PAPER:

Dynamic financial processes identification using sparse autoregressive reservoir computers

AUTHORS

Fredy Vides
Idelfonso B. R. Nogueira

July 6, 2023

Dynamic financial processes identification using sparse autoregressive reservoir computers

Fredy Vides[1, 2, a)] and Idelfonso B. R. Nogueira[3, b)]
[1)]*Department of Statistics and Research, National Commission of Banks and Insurance Companies*
[2)]*Department of Applied Mathematics, School of Mathematics and Computer Science, Universidad Nacional Autónoma de Honduras, Tegucigalpa, Honduras*
[3)]*Department of Chemical Engineering, Norwegian University of Science and Technology, Norway*

(Dated: July 6, 2023)

In this document, some general results in structured matrix approximation theory with applications to autoregressive representation of dynamic financial processes are presented. Firstly, a generic nonlinear time delay embedding is considered for the time series data sampled from a financial or economic system under study. Secondly, sparse least-squares and structured matrix approximation methods are applied to identify approximate representations of the output coupling matrices, determining the autoregressive representations of the recursive models corresponding to some given financial system under consideration. Prototypical algorithms based on the aforementioned techniques, together with some applications to approximate identification and predictive simulation of dynamic nonlinear financial and economic systems that may or may not exhibit chaotic behavior are presented.

Dynamic financial processes are often described by complex systems that exhibit dynamic behavior, making it challenging to model and predict them accurately. However, combining sparse representation techniques with nonlinear autoregressive reservoir computers (NARCs) can provide significant advantages for financial processes dynamics modeling. The first advantage of this approach is its ability to capture the complex and nonlinear dynamics of financial processes data. NARCs are well-suited for modeling complex relationships between input and output data, and by combining them with sparse representation techniques, it is possible to identify the key dynamic components of the process. This allows for more accurate and precise modeling of the underlying dynamics of financial processes. The second advantage of this approach is that it enables more efficient use of data. NARCs can learn from a relatively small amount of data, making them ideal for financial processes data, which is often limited in scope and complexity. By identifying the most important variables or components in a financial process, it is possible to focus on the most relevant data, which can improve the efficiency of the modeling process and save time and resources. The third advantage of this approach is its flexibility and adaptability. NARCs can adapt to changing conditions and respond quickly to new data inputs, making them ideal for financial processes that are constantly evolving. By combining NARCs with sparse representation techniques, it is possible to identify changes in the underlying structure of a financial process and adjust the model accordingly. In conclusion, combining sparse representation techniques with time series models based on nonlinear autoregressive reservoir computers provides several advantages for financial processes dynamics modeling. It enables more accurate modeling of complex and nonlinear dynamics, more efficient use of data, and greater flexibility and adaptability to changing conditions.

I. INTRODUCTION

Recursive models and reservoir computers provide powerful computational tools for financial and economic system identification and simulation[1]. Recently, some novel architectures named next generation reservoir computers[4] have been devised. This document presents some elements of the theory and algorithmics corresponding to the computation of some particular types of autoregressive reservoir computers. The study reported in this document is focused on reservoir computers whose architecture can be approximately represented by either linear or nonlinear autoregressive vector models.

The main contribution of the work reported in this document is the application of *colaborative schemes* involving structured matrix approximation methods, together with linear and nonlinear autoregressive models, to the simulation of dynamic financial processes. Some theoretical aspects of the aforementioned methods are described in §III. As a byproduct of the work reported in this document, a toolset of Matlab programs for semilinear sparse signal model computation based on the ideas presented in §III and §IV has been developed and is available in[8].

Even though, the applications of the structure preserving function approximation technology developed as part of the work reported in this document can range from numerical modeling of cyber-physical systems[9], to climate simulation[6]. We will focus on applications to financial processes identification in this paper.

Financial processes have become a complex system where several dynamic entities constantly communicate and affect each other. Hence, the financial processes identification has become a critical aspect of modern finance. The identified models can become a helpful tool for institutions to analyze

[a)]Electronic mail: fredy.vides@cnbs.gob.hn
[b)]Electronic mail: idelfonso.b.d.r.nogueira@ntnu.no

and predict financial trends, manage risk, and make informed investment decisions (Bodie et al., 2014). However, the complexity and uncertainty of financial markets make these tasks challenging. Financial processes often exhibit nonlinear and complex behavior, which makes it difficult to model and identify the underlying dynamics (Cont, 2001). Traditional linear models may fail to capture the intricate relationships between variables, leading to inaccurate predictions and suboptimal decision-making.

Despite the challenges posed by the factors described above, data quality, and market efficiency, machine learning techniques offer promising solutions for improving the accuracy and utility of financial models. Machine learning has been used to identify the relationship between the key financial ratios that characterize a firm's financial position. For instance, Dixon, Klabjan, and Bang's[2] work applies deep learning to predict financial market movements. The authors use a classification approach to predict financial market movements. Their findings suggest that deep learning algorithms can provide valuable insights and predictions about financial market movements, outperforming traditional methods. Sirignano and Cont[7] propose a deep learning model to identify the dynamics of price formation of a high-frequency limit order book. Their model was able to capture universal features of price formation across different markets, highlighting the potential of machine learning to model complex financial systems.

Overall, the recent literature suggests that machine learning has significant potential in modeling financial data. These techniques are increasingly utilized to capture complex patterns, make accurate predictions, and optimize decision-making in the financial domain. However, it is still an open issue to be investigated. In this scenario, this work also contributes to the field of financial data identification by applying the proposed tools in this context leading to a better understanding of the underlying financial processes addressed here.

A prototypical algorithm for the computation of sparse structured recursive models based on the ideas presented in §III, is presented in §IV. Some numerical simulations of financial processes based on the prototypical algorithm presented in §IV are documented in §V.

II. PRELIMINARIES AND NOTATION

The symbols $\mathbb{R}^+$ and $\mathbb{Z}^+$ will be used to denote the positive real numbers and positive integers, respectively. For any pair $p,n \in \mathbb{Z}^+$ the expression $d_p(n)$ will denote the posirive integer $d_p(n) = n(n^p-1)/(n-1)+1$. For any finite group G, the expression G will be used to denote the order of G, that is, the number of elements of G. Given $\delta > 0$, let us consider the function defined by the expression

$$H_\delta(x) = \begin{cases} 1, & x > \delta \\ 0, & x \leq \delta \end{cases}.$$

Given a matrix $A \in \mathbb{C}^{m\times n}$ with singular values[5] (§2.5.3) denoted by the expressions $s_j(A)$ for $j = 1,\ldots,\min\{m,n\}$. We will write $\mathrm{rk}_\delta(A)$ to denote the number

$$\mathrm{rk}_\delta(A) = \sum_{j=1}^{\min\{m,n\}} H_\delta(s_j(A)).$$

For a nonzero matrix $A \in \mathbb{R}^{m\times n}$, the symbol A^+ will be used to denote the pseudoinverse[5] (§5.5.4) of A.

Given a scalar time series $\Sigma = \{x_t\}_{t\geq 1} \subset \mathbb{R}^\kappa$, a positive integer L and any $t \geq L$, we will write $\mathbf{x}_L(t)$ to denote the vector

$$\mathbf{x}_L(t) = \begin{bmatrix} \mathbf{x}_L^{(1)}(t)^\top & \mathbf{x}_L^{(2)}(t)^\top & \cdots & \mathbf{x}_L^{(n)}(t)^\top \end{bmatrix}^\top \in \mathbb{R}^{nL},$$

with

$$\mathbf{x}_L^{(j)}(t) = \begin{bmatrix} x_{t-L+1}^{(j)} & x_{t-L+2}^{(j)} & \cdots & x_{t-1}^{(j)} & x_t^{(j)} \end{bmatrix}^\top \in \mathbb{R}^L.$$

for $1 \leq j \leq n$, where $x_{j,s}$ denotes the scalar j-component of each element x_s in the vector time series Σ, for $s \geq 1$.

The identity matrix in $\mathbb{R}^{n\times n}$ will be denoted by I_n, and we will write $\hat{e}_{j,n}$ to denote the matrices in $\mathbb{R}^{n\times 1}$ representing the canonical basis of $\mathbb{R}^n$ (each $\hat{e}_{j,n}$ corresponds to the j-column of I_n). For any vector $x \in \mathbb{R}^n$, we will write $\|x\|$ to denote the Euclidean norm of x. Given a matrix $X \in \mathbb{R}^{m\times n}$, the expression $\|X\|_F$ will denote the Frobenius norm of X.

For any integer $n > 0$, in this article, we will identify the vectors in $\mathbb{R}^n$ with column matrices in $\mathbb{R}^{n\times 1}$.

Given two matrices $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{p\times q}$, the tensor Kronecker tensor product $A \otimes B \in \mathbb{R}^{mp\times nq}$ is determined by the following operation.

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}$$

For any integer $p > 0$ and any matrix $X \in \mathbb{R}^{m\times n}$, we will write $X^{\otimes p}$ to denote the operation determined by the following expression.

$$X^{\otimes p} = \begin{cases} X & ,p=1 \\ X \otimes X^{\otimes(p-1)} & ,p \geq 2 \end{cases}$$

We will also use the symbol Π_p to denote the operator $\Pi_p : \mathbb{R}^n \to \mathbb{R}^{n^p}$ that is determined by the expression $\Pi_p(x) := x^{\otimes p}$, for each $x \in \mathbb{R}^n$.

For any matrix $A \in \mathbb{R}^{m\times n}$, we will denote by $\mathrm{colsp}(A)$ the columns space of the matrix A. Given a list $A_1, A_2, \ldots, A_m$ such that for $1 \leq j \leq m$, $A_j \in \mathbb{R}^{n_j\times n_j}$ for some integer $n_j > 0$. The expression $A_1 \oplus A_2 \oplus \cdots \oplus A_m$ will denote the block diagonal matrix

$$A_1 \oplus A_2 \oplus \cdots \oplus A_m = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_m \end{bmatrix},$$

where the zero matrix blocks have been omitted.

We will write $\mathbf{S}^1$ to denote the set $\{z \in \mathbb{C} : |z| = 1\}$. Given any matrix $X \in \mathbb{R}^{m\times n}$, we will write $X^\top$ to denote the transpose $X^\top \in \mathbb{R}^{n\times m}$ of X. A matrix $P \in \mathbb{C}^{n\times n}$ will be called an orthogonal projector whenever $P^2 = P = P^\top$. The group of orthognal matrices in $\mathrm{R}^{n\times n}$ will be denoted by $\mathbb{O}(n)$ in this study. Given any matrix $A \in \mathbb{R}^{n\times n}$, we will write $\Lambda(A)$ to denote the spectrum of A, that is, the set of eigenvalues of A.

III. STRUCTURED RECURSIVE MODEL IDENTIFICATION

A. Low-rank approximation and sparse linear least squares solvers

In this section some low-rank approximation methods with applications to the solution of sparse linear least squares problems are presented.

Definition III.1. *Given $\delta > 0$ and a matrix $A \in \mathbb{C}^{m\times n}$, we will write $\mathrm{rk}_\delta(A)$ to denote the nonnegative integer determined by the expression*

$$\mathrm{rk}_\delta(A) = \sum_{j=1}^{\min\{m,n\}} H_\delta(s_j(A)),$$

where the numbers $s_j(A)$ represent the singular values corresponding to an economy-sized singular value decomposition of the matrix A.

Lemma III.2. *We will have that $\mathrm{rk}_\delta\left(A^\top\right) = \mathrm{rk}_\delta(A)$ for each $\delta > 0$ and each $A \in \mathbb{C}^{m\times n}$.*

Proof. Given an economy-sized singular value decomposition

$$U \begin{bmatrix} s_1(A) & & & \\ & s_2(A) & & \\ & & \ddots & \\ & & & s_{\min\{m,n\}}(A) \end{bmatrix} V = A$$

we will have that

$$V^\top \begin{bmatrix} s_1(A) & & & \\ & s_2(A) & & \\ & & \ddots & \\ & & & s_{\min\{m,n\}}(A) \end{bmatrix} U^\top = A^\top$$

is an economy-sized singular value decomposition of $A^\top$. This implies that

$$\mathrm{rk}_\delta\left(A^\top\right) = \sum_{j=1}^{\min\{m,n\}} H_\delta(s_j(A)) = \mathrm{rk}_\delta(A)$$

and this completes the proof. ☐

Lemma III.3. *Given $\delta > 0$ and $A \in \mathbb{C}^{m\times n}$ we will have that $\mathrm{rk}_\delta(A) \leq \mathrm{rk}(A)$.*

Proof. We will have that $\mathrm{rk}(A) = \sum_{j=1}^{\min\{m,n\}} H_0(s_j(A)) \geq \sum_{j=1}^{\min\{m,n\}} H_\delta(s_j(A)) = \mathrm{rk}_\delta(A)$. This completes the proof. ☐

Theorem III.4. *Given $\delta > 0$ and $y, x_1, \ldots, x_m \in \mathbb{C}^n$, let*

$$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_m \\ | & | & & | \end{bmatrix}.$$

If $\mathrm{rk}_\delta(X) > 0$ and if we set $r = \mathrm{rk}_\delta(X)$ and $s_{n,m}(r) = \sqrt{r(\min\{m,n\} - r)}$ then, there are a rank r orthogonal projector Q, r vectors $x_{j_1}, \ldots, x_{j_r} \in \{x_1, \ldots, x_m\}$ and r scalars $c_1 \ldots, c_r \in \mathbb{C}$ such that $\|X - QX\|_F \leq (s_{n,m}(r)/\sqrt{r})\delta$, and $\|y - \sum_{k=1}^r c_k x_{j_k}\| \leq \left(\sum_{k=1}^r |c_k|^2\right)^{\frac{1}{2}} s_{n,m}(r)\delta + \|(I_n - Q)y\|$.

Proof. Let us consider an economy-sized singular value decomposition $USV = A$. If u_j denotes the j-column of U, let Q be the rank $r = \mathrm{rk}_\delta(A)$ orthogonal projector determined by the expression $Q = \sum_{j=1}^r u_j u_j^*$. It can be seen that

$$\begin{aligned} \|X - QX\|_F^2 &= \sum_{j=r+1}^{\min\{m,n\}} s_j(X)^2 \\ &\leq (\min\{m,n\} - r)\delta^2 = \frac{s_{n,m}(r)^2}{r}\delta^2. \end{aligned}$$

Consequently, $\|X - QX\|_F \leq \frac{s_{n,m}(r)}{\sqrt{r}}\delta$.

Let us set.

$$\hat{X} = \begin{bmatrix} | & | & & | \\ \hat{x}_1 & \hat{x}_2 & \cdots & \hat{x}_m \\ | & | & & | \end{bmatrix} = QX$$

$$\hat{X}_y = \begin{bmatrix} | & | & & | & | \\ \hat{x}_1 & \hat{x}_2 & \cdots & \hat{x}_m & \hat{y} \\ | & | & & | & | \end{bmatrix} = Q\begin{bmatrix} X & y \end{bmatrix}$$

Since by lemma III.3 $\mathrm{rk}(X) \geq \mathrm{rk}_\delta(X)$, we will have that $\mathrm{rk}(\hat{X}) = r = \mathrm{rk}_\delta(X) > 0$, and since we also have that $\hat{x}_1, \ldots, \hat{x}_m, \hat{y} \in \mathrm{span}(\{u_1, \ldots, u_r\})$, there are r linearly independent $\hat{x}_{j_1}, \ldots, \hat{x}_{j_r} \in \{\hat{x}_1, \ldots, \hat{x}_m\}$ such that $\mathrm{span}(\{u_1, \ldots, u_r\}) = \mathrm{span}(\{\hat{x}_{j_1}, \ldots, \hat{x}_{j_r}\})$, this in turn implies that $\hat{y} \in \mathrm{span}(\{\hat{x}_{j_1}, \ldots, \hat{x}_{j_r}\})$ and there are $c_1, \ldots, c_r \in \mathbb{C}$ such that $\hat{y} = \sum_{k=1}^r c_k \hat{x}_{j_k}$. It can be seen that for each $z \in \{x_1, \ldots, x_m\}$

$$\|z - Qz\| \leq \|X - QX\|_F \leq \frac{s_{n,m}(r)}{\sqrt{r}}\delta,$$

and this in turn implies that

$$\begin{aligned} \left\|y - \sum_{k=1}^r c_k x_{j_k}\right\| &= \left\|y - \sum_{k=1}^r c_k x_{j_k} - \left(\hat{y} - \sum_{k=1}^r c_k \hat{x}_{j_k}\right)\right\| \\ &= \left\|y - \sum_{k=1}^r c_k x_{j_k} - Q\left(y - \sum_{k=1}^r c_k x_{j_k}\right)\right\| \\ &\leq \left(\sum_{k=1}^r |c_k|^2\right)^{\frac{1}{2}} s_{n,m}(r)\delta + \|(I_n - Q)y\|. \end{aligned}$$

This completes the proof. ☐

As a direct implication of theorem III.4 one can obtain the following corollary.

Corollary III.5. *Given $\delta > 0$, $A \in \mathbb{C}^{m\times n}$ and $y \in \mathbb{C}^m$. If $\mathrm{rk}_\delta(A) > 0$ and if we set $r = \mathrm{rk}_\delta(A)$ and $s_{n,m}(r) = \sqrt{r(\min\{m,n\}-r)}$ then, there are $x \in \mathbb{C}^n$ and a rank r orthogonal projector Q that does not depend on y, such that $\|Ax-y\| \leq \|x\|s_{n,m}(r)\delta + \|(I_m-Q)y\|$ and x has at most r nonzero entries.*

Proof. Let us set $x = \mathbf{0}_{n,1}$ and $a_j = A\hat{e}_{j,n}$ for $j = 1,\ldots,n$. Since $r = \mathrm{rk}_\delta(A) > 0$ and $s_{n,m}(r) = \sqrt{r(\min\{m,n\}-r)}$, by theorem III.4 we will have that there is a rank r orthogonal projector Q such that $\|A-QA\|_F \leq (s_{n,m}(r)/\sqrt{r})\delta$, and without loss of generality r vectors $a_{j_1},\ldots,a_{j_r} \in \{a_1,\ldots,a_n\}$ and r scalars $c_1\ldots,c_r \in \mathbb{C}$ with $j_1 \leq j_2 \leq \cdots \leq j_r$ (reordering the indices j_k if necessary), such that $\|y - \sum_{k=1}^r c_k a_{j_k}\| \leq \left(\sum_{k=1}^r |c_k|^2\right)^{\frac{1}{2}} s_{n,m}(r)\delta + \|(I_m-Q)y\|$. If we set $x_{j_k} = c_k$ for $k = 1,\ldots,r$, we will have that $\|x\| = \left(\sum_{k=1}^r |c_k|^2\right)^{\frac{1}{2}}$ and $Ax = \sum_{k=1}^r x_{j_k}a_{j_k} = \sum_{k=1}^r c_k a_{j_k}$. Consequently, $\|Ax-y\| \leq \|x\|s_{n,m}(r)\delta + \|(I_m-Q)y\|$. This completes the proof. □

The results and ideas presented in this section can be translated into a sparse linear least squares solver algorithm described by algorithm A.1 in §IV.

B. Sparse structured nonlinear autoregressive model identification

consider the time series data $\Sigma \subset \mathbb{R}^n$ corresponding to an orbit determined by the difference equation

$$x_{t+1} = \mathscr{T}(x_t), \tag{III.1}$$

for some discrete-time dynamic financial model $(\hat{\mathbf{\Sigma}}, \mathscr{T})$ to be identified. One may need to be preprocess the time series data before proceeding with the approximate representation of a suitable evolution operator. For this purpose, given some prescribed integer $L > 0$, one can consider the time series $\mathscr{D}_L(\Sigma)$ determined by the expression.

$$\mathscr{D}_L(\Sigma) = \{\mathbf{x}_L(t)\}_{t\geq L}$$

For the dilated time series $\mathscr{D}_L(\Sigma)$, the previously considered recurrence relation $x_{t+1} = \mathscr{T}(x_t), t \geq 1$, induces the following difference equations

$$\mathbf{x}_L(t+1) = \tilde{\mathscr{T}}(\mathbf{x}_L(t)), \tag{III.2}$$

for $t \geq L$. Where $\tilde{\mathscr{T}}$ is some evolution operator to be identified.

For any $p \geq 1$, let us consider the map $\eth_p : \mathbb{R}^n \to \mathbb{R}^{d_p(n)}$ for $d_p(n) = n(n^p-1)/(n-1)+1$, that is determined by the expression.

$$\eth_p(x) := \begin{bmatrix} \Pi_1(x) \\ \Pi_2(x) \\ \vdots \\ \Pi_p(x) \\ 1 \end{bmatrix} = \begin{bmatrix} x^{\otimes 1} \\ x^{\otimes 2} \\ \vdots \\ x^{\otimes p} \\ 1 \end{bmatrix}$$

Given integers $p, L > 0$, an orbit $\Sigma = \{x_t\}_{t\geq 1} \subset \mathbb{R}^n$ of a dynamic financial process. For a finite sample $\Sigma_N = \{x_t\}_{t=1}^T \subset \Sigma$, let us consider the matrices:

$$\begin{aligned} \mathbf{H}_L^{(0,p)}(\Sigma_T) &= \begin{bmatrix} \eth_p(\mathbf{x}_L(L)) & \cdots & \eth_p(\mathbf{x}_L(T-1)) \end{bmatrix} \\ \mathbf{H}_L^{(1)}(\Sigma_T) &= \begin{bmatrix} \mathbf{x}_L(L+1) & \cdots & \mathbf{x}_L(T) \end{bmatrix} \end{aligned} \tag{III.3}$$

The operator identification mechanism used in this study for dilated systems of the form (III.2), will be described by the expression:

$$\hat{\mathscr{T}}(\mathbf{x}_L(t)) = W\eth_p(\mathbf{x}_L(t)),\ t \geq L, \tag{III.4}$$

for some matrix $W \in \mathbb{R}^{n\times d_p(n)}$ to be determined, with $d_p(n) = n(n^p-1)/(n-1)+1$. Applying by the operator theoretic techniques and ideas previously presented in this section, the matrix W in (III.4) can be estimated by approximately solving the matrix equation

$$W\mathbf{H}_L^{(0,p)}(\Sigma_T) = \mathbf{H}_L^{(1)}(\Sigma_T). \tag{III.5}$$

The devices described by (III.4) are called autoregressive reservoir computers (ARC) in this paper.

For any given integers $L, n, p > 0$. Taking advantage of the maps $\eth_p$, one can find a integer $0 < r_p(n) < d_p(n)$ together with a sparse matrix $R_{p,L}(n) \in \mathbb{R}^{r_p(n)\times d_p(n)}$, such that $R_{p,L}(n)^+ R_{p,L}(n)\eth_p(x) \approx \eth_p(x)$ for $x \in \mathbb{R}^{nL}$. The existence of the pair $r_p(n), R_{p,L}(n)$ is determined by the following theorem.

Theorem III.6. *Given $\varepsilon \in \mathbb{R}^+$ and $L, n, p \in \mathbb{Z}^+$. There are an integer $0 < \rho_p(n) < d_p(n)$ and a sparse matrix $R_{p,L}(n) \in \mathbb{R}^{\rho_p(n)\times d_p(n)}$ with $d_p(n)$ nonzero entries, such that $\|R_{p,L}(n)^+ R_{p,L}(n)\eth_p(x) - \eth_p(x)\| \leq \sqrt{d_p(nL)}\varepsilon$ for each $x \in \mathbb{R}^{nL}$.*

Proof. Let us consider the symmetric group $\mathfrak{S}_{nL-1}$ on $nL-1$ letters, and let us consider any finite set of distinct points $\{\hat{x}_1,...,\hat{x}_{nL},\hat{x}_{nL+1}\} \subset \mathbb{R}$ such that for each $1 \leq m \leq p$ and every pair on index sets $\{i_1,\ldots,i_m\},\{j_1,\ldots,j_m\} \subset \{1,\ldots,nL+1\}$:

$$\prod_{k=1}^m \hat{x}_{j_k} = \prod_{k=1}^m \hat{x}_{i_k}, \Leftrightarrow \exists\sigma \in \mathfrak{S}_{nL-1} : i_k = \sigma(j_k), \forall 1 \leq k \leq m. \tag{III.6}$$

That is, the previously considered products coincide if and only if, the index set $\{i_1,\ldots,i_m\}$ is a permutation of the set $\{j_1,\ldots,j_m\}$, for each $1 \leq m \leq p$.

Let us now set $\mathbf{y} = \nu \begin{bmatrix} \hat{x}_1 & \hat{x}_2 & \cdots & \hat{x}_{nL} \end{bmatrix}^\top$, $d = d_p(n)$ and

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x}_1 & \cdots & \tilde{x}_d \end{bmatrix}^\top := \eth_p(\mathbf{y}).$$

If in addition we consider the assignments:

$$\tilde{x}_d := \hat{x}_{nL+1},$$

$$R = e_{1,d}^\top.$$

Then, for each $j = 2,\ldots,d$ one can find $1 \leq k_1(j),\ldots,k_{n_j}(j) \leq d$ such that $|\tilde{x}_j - \tilde{x}_{k_m(j)}| \leq \varepsilon$, for every

$1 \leq m \leq n_j$. Consequently, if we set $R_0 := (1/n_j)\sum_{l=1}^{n_j} \hat{e}_{l,d}^T$ and

$$R := \begin{bmatrix} R \\ R_0 \end{bmatrix},$$

whenever $k_1(j) = j$. After iterating on the previously described procedure for $2 \leq j \leq d$, we can define $R_{p,L}(n) := R$ and we can set the value of $\rho_p(n)$ as the number of rows of $R_{p,L}(n)$.

Given $x \in \mathbb{R}^{nL}$. Based on the structure of $R_{p,L}(n)$ determined by the constructive procedure used for its computation, it can be verified that $R_{p,L}(n)^+$ is well defined and in addition:

$$\|R_{p,L}(n)^+ R_{p,L}(n)\eth_p(x) - \eth_p(x)\| \leq \sqrt{d_p(n)}\varepsilon.$$

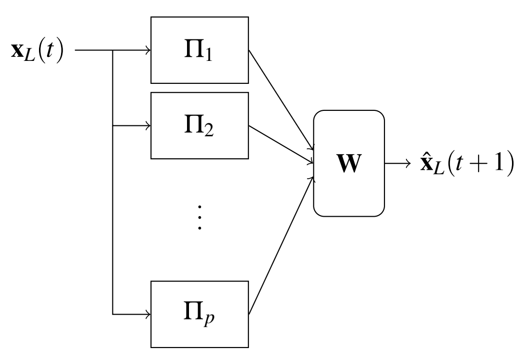
This completes the proof. □

In order to reduce to computational effort corresponding to the solution of (III.5), using the matrix $R_{p,L}(n)$ described by Theorem III.6, one can obtain an approximate reduced representation of (III.5) determined by the expression.

$$\bar{W} R_{p,L}(n)\mathbf{H}_L^{(0,p)}(\Sigma_T) = \mathbf{H}_L^{(1)}(\Sigma_T) \tag{III.7}$$

The architecture of the autoregressive reservoir computers considered in this study was inspired by next generation reservoir computers[4].

Schematically, the autoregressive models considered in this study can be described by a block diagram of the form,



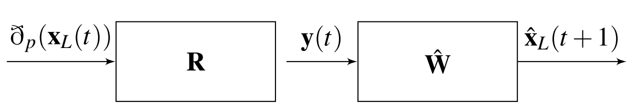
(III.8)

where for each $t \geq L$, the block $\mathbf{W}$ is determined by the expression

$$\begin{aligned}\mathbf{W}(\Pi_1(\mathbf{x}_L(t)),\ldots,\Pi_p(\mathbf{x}_L(t))) &:= \tilde{W}\begin{bmatrix}\Pi_1(\mathbf{x}_L(t)) \\ \vdots \\ \Pi_p(\mathbf{x}_L(t))\end{bmatrix} + c_W \\ &= \begin{bmatrix}\tilde{W} & c_W\end{bmatrix}\eth_p(\mathbf{x}_L(t))\end{aligned}$$

and where the matrix $W = \begin{bmatrix}\tilde{W} & c_W\end{bmatrix}$ determined by (III.5).

The structure of the generic block $\mathbf{W}$ ub (III.8) can be factoted in the form



(III.9)

The layers $\mathbf{R}$ and $\hat{\mathbf{W}}$ of the device (III.9) are determined by the expressions

$$\mathbf{R}(\mathbf{x}) = \hat{R}\mathbf{x},$$
$$\hat{\mathbf{W}}(\mathbf{y}) = W\hat{R}^+\mathbf{y}$$

for any pair of suitable vectors $\mathbf{x},\mathbf{y}$. Where W is a sparse representation of an approximate solution (III.5) and $\hat{R}$ is determined by Theorem III.6.

Using some suitable training subset Σ_S of a given data sample Σ_T from an arbitrary signal data $\Sigma = \{x_t\}_{t\geq 1} \subset \mathbb{R}^n$ under consideration, the parameters of the block $\mathbf{W}$ of (III.8) are fitted using Σ_S. Simultaneously, a sparse representation of the matrix parameters corresponding to the block $\mathbf{W}$ of the resulting model is computed.

Using the reservoir computer models described by (III.4), (III.8) and (III.9), we can compute approximate representations of evolution operators that satisty (III.2) using the expression

$$\hat{\mathscr{T}}(\mathbf{x}_L(t)) := \hat{K}\left(\hat{\mathbf{W}} \circ \mathbf{R} \circ \eth_p(\mathbf{x}_L(t)) = \hat{K}W\eth_p(\mathbf{x}_L(t))\right), \tag{III.10}$$

for each $t \geq L$. Furthermore, we can use the identified RRC model $\hat{\mathscr{T}}$ to simulate the behavior $x_{t+1} = \mathscr{T}(x_t)$ of the system described by (III.1) for $L \leq t \leq \tau$, by peforming the operation:

$$\mathbf{T}(\mathbf{x}_L(t)) := \hat{K}\hat{\mathscr{T}}(\mathbf{x}_L(t)) = \hat{K}W\eth_p(\mathbf{x}_L(t))), \tag{III.11}$$

for some $\tau > 0$ with

$$\hat{K} = \begin{bmatrix} \hat{e}_{1,nL}^\top \\ \hat{e}_{L+1,nL}^\top \\ \vdots \\ \hat{e}_{(n-1)L+1,nL}^\top \end{bmatrix}.$$

In this article, we will use the following notion of sparse representation. Given $\delta > 0$ and two matrices $A \in \mathbb{R}^{m\times n}$ and $X \in \mathbb{R}^{n\times p}$, a matrix $\hat{X} \in \mathbb{R}^{n\times p}$ is an approximate sparse representation of X, or a sparse representation of X for short, if $\|\hat{X}A - XA\|_F \leq C\delta$ for some $C > 0$ that does not depend on δ, and $\hat{X}$ has fewer nonzero entries than X.

Theorem III.7. *Given $\delta > 0$, two integers $p,L > 0$, a finite group G, a sample $\Sigma_T = \{x_t\}_{t=1}^T$ from a financial modeling's orbit $\Sigma = \{x_t\}_{t\geq 1} \subset \mathbb{R}^n$ with $T > L$, and a matrix solvent $\bar{W} \in \mathbb{R}^{nL\times r_p(nL)}$ of* (III.7) *with $R_{p,L}(n)$ and $r_p(n)$ determined by Theorem III.6. If $r = \mathrm{rk}_\delta(R_{p,L}(n)\mathbf{H}_L^{(0,p)}(\Sigma_T)) > 0$, then there is a sparse matrix $\hat{W} \in \mathbb{R}^{nL\times \rho_p(nL)}$ with at most $r\rho_p(nL)$ nonzero entries such that*

$$\|\hat{W}R_{p,L}(n)\mathbf{H}_L^{(0,p)}(\Sigma_T) - \bar{W}R_{p,L}(n)\mathbf{H}_L^{(0,p)}(\Sigma_T)\|_F \leq K\delta, \tag{III.12}$$

for $K = \sqrt{nL(\min\{\rho_p(nL), T-L\} - r)}(\sqrt{r}\|\hat{W}\|_F + \|\bar{W}\|_F)$, where $\rho_p(nL)$ is the integer described by Theorem III.6.

Proof. Let us set $H = R_{p,L}(n)\mathbf{H}_{L,G}^{(0,p)}(\Sigma_T)^\top$ and $Y = H\bar{W}^\top$. It suffices to prove that there is a sparse matrix $\hat{W} \in \mathbb{R}^{nL\times \rho_p(nL)}$ with at most $r\rho_p(nL)$ nonzero entries such that

$$\|H\hat{W}^\top - Y\|_F \leq K\delta.$$

Since we have that

$$\mathrm{rk}_\delta(H) = \mathrm{rk}_\delta\left(\left(R_{p,L}(n)\mathbf{H}_{L,G}^{(0,p)}(\Sigma_T)\right)^\top\right) = \mathrm{rk}_\delta(R_{p,L}(n)\mathbf{H}_{L,G}^{(0,p)}(\Sigma_T)) > 0$$

by Lemma III.2. By Corollary III.5, if we set $r = \mathrm{rk}_\delta(H)$ and $\alpha = \sqrt{r(\min\{\rho_p(nL), T-L\} - r)}$. We will have that there is a rank r orthogonal projector Q such that for each $j = 1, \ldots, nL$, there is $\hat{v}_j \in \mathbb{R}^{nL}$ with at most r nonzero entries, for which $\|H\hat{v}_j - Y\hat{e}_{j,M}\| \leq \alpha\|\hat{v}_j\|\delta + \|(I_{T-L} - Q)Y\hat{e}_{j,nL}\|$. Consequently, if we set

$$\hat{W} = \begin{bmatrix} | & & | \\ \hat{v}_1 & \cdots & \hat{v}_{nL} \\ | & & | \end{bmatrix}^\top$$

we will have that $\hat{W}$ has at most nrL nonzero entries and

$$\begin{aligned}\|H\hat{W}^\top - Y\|_F^2 &= \sum_{j=1}^{nL} \|H\hat{v}_j - Y\hat{e}_{j,nL}\|^2 \\ &\leq M(\alpha\|\hat{W}\|_F\delta + \|(I_{T-L} - Q)Y\|_F)^2,\end{aligned}$$

and this in turn implies that,

$$\|H\hat{W}^\top - Y\|_F \leq \sqrt{nL}(\alpha\|\hat{W}\|_F\delta + \|(I_{T-L} - Q)H\|_F\|\bar{W}\|_F). \tag{III.13}$$

By (III.13) and by Theorem III.4 we will have that

$$\begin{aligned}\|H\hat{W}^\top - Y\|_F &\leq \sqrt{nL}(\alpha\|\hat{W}\|_F\delta + (\alpha/\sqrt{r})\|\bar{W}\|_F\delta) \\ &= \alpha\sqrt{(nL/r)}(\sqrt{r}\|\hat{A}\|_F + \|A\|_F)\delta = K\delta.\end{aligned}$$

This completes the proof. □

IV. ALGORITHMS

The sparse model identification methods presented in §III A can be translated into prototypical algorithms that will be presented in this section, some programs for data reading and writing, synthetic signals generation, and predictive simulation are also include as part of the **SPORT** tool-set available in? .

A. Sparse linear least squares solver and structured assembling matrix identification algorithms

As an application of the results and ideas presented in §III A one can obtain a prototypical sparse linear least squares solver algorithm like algorithm A.1.

The least squares problems $c = \arg\min_{\hat{c}\in\mathbb{C}^K}\|\hat{A}\hat{c} - y\|$ to be solved as part of the process corresponding to algorithm A.1 can be solved with any efficient least squares solver available in the language or program where the sparse linear least squares solver algorithm is implemented. For the Matlab and Julia implementations of algorithm A.1 written as part of this research project the `backslash "\"` operator is used, and for the Python version of algorithm A.1 the function `lstsq` is implemented.

Algorithm A.1 SLRSolver: Sparse linear least squares solver algorithm

Data: $A \in \mathbb{C}^{m\times n}$, $Y \in \mathbb{C}^{m\times p}$, $\delta > 0$, $N \in \mathbb{Z}^+$, $\varepsilon > 0$
Result: $X = \mathbf{SLRSolver}(A, Y, \delta, N, \varepsilon)$

1. Compute economy-sized SVD $USV = A$
2. Set $s = \min\{m, n\}$
3. Set $r = \mathrm{rk}_\delta(A)$
4. Set $U_\delta = \sum_{j=1}^r U\hat{e}_{j,s}\hat{e}_{j,s}^*$
5. Set $T_\delta = \sum_{j=1}^r(\hat{e}_{j,s}^*S\hat{e}_{j,s})^{-1}\hat{e}_{j,s}\hat{e}_{j,s}^*$
6. Set $V_\delta = \sum_{j=1}^r \hat{e}_{j,s}\hat{e}_{j,s}^*V$
7. Set $\hat{A} = U_\delta^* A$
8. Set $\hat{Y} = U_\delta^* Y$
9. Set $X_0 = V_\delta^* T_\delta \hat{Y}$
10. **for** $j = 1, \ldots, p$ **do**
    - Set $K = 1$
    - Set error $= 1 + \delta$
    - Set $c = X_0\hat{e}_{j,p}$
    - Set $x_0 = c$
    - Set $\hat{c} = \begin{bmatrix}\hat{c}_1 & \cdots & \hat{c}_n\end{bmatrix}^\top = \begin{bmatrix}|\hat{e}_{1,n}^*c| & \cdots & |\hat{e}_{n,n}^*c|\end{bmatrix}^\top$
    - Compute permutation $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that: $\hat{c}_{\sigma(1)} \geq \hat{c}_{\sigma(2)} \geq \cdots \geq \hat{c}_{\sigma(n)}$
    - Set $N_0 = \max\left\{\sum_{j=1}^n H_\varepsilon\left(\hat{c}_{\sigma(j)}\right), 1\right\}$
    - **while** $K \leq N$ **and** error $> \delta$ **do**
        - Set $x = \mathbf{0}_{n,1}$
        - Set $A_0 = \sum_{j=1}^{N_0}\hat{A}\hat{e}_{\sigma(j),n}\hat{e}_{j,N_0}^*$
        - Solve $c = \arg\min_{\tilde{c}\in\mathbb{C}^{N_0}}\|A_0\tilde{c} - \hat{Y}\hat{e}_{j,p}\|$
        - **for** $k = 1, \ldots, N_0$ **do**
            - Set $x_{\sigma(k)} = \hat{e}_{k,N_0}^* c$
        - **end for**
        - Set error $= \|x - x_0\|_\infty$
        - Set $x_0 = x$
        - Set $\hat{c} = \begin{bmatrix}\hat{c}_1 & \cdots & \hat{c}_n\end{bmatrix}^\top = \begin{bmatrix}|\hat{e}_{1,n}^*x| & \cdots & |\hat{e}_{n,n}^*x|\end{bmatrix}^\top$
        - Compute permutation $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that: $\hat{c}_{\sigma(1)} \geq \hat{c}_{\sigma(2)} \geq \cdots \geq \hat{c}_{\sigma(n)}$
        - Set $N_0 = \max\left\{\sum_{j=1}^n H_\varepsilon\left(\hat{c}_{\sigma(j)}\right), 1\right\}$
        - Set $K = K + 1$
    - **end while**
    - Set $x_j = x$
11. **end for**
12. Set $X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_p \\ | & | & & | \end{bmatrix}$

return X

In this section we focus on the applications of the structured matrix approximation methods presented in SIII, to dynamical financial systems identification via autoregressive reservoir computers.

Algorithm A.2 Compression matrix computation algorithm

Data: $n,p,L \in \mathbb{Z}^+$, $\nu,\varepsilon \in \mathbb{R}^+$.
Result: COMPRESSION MATRIX FACTOR: $R_{p,L}(n)$

1. Choose nL pseudorandom numbers $\hat{x}_1,\ldots,\hat{x}_{nL} \in \mathbb{R}$ from $N(0,1)$
2. Set $\mathbf{y} = \nu \begin{bmatrix} \hat{x}_1 & \hat{x}_2 & \cdots & \hat{x}_{nL} \end{bmatrix}^\top$
3. Set $d = d_p(n)$
4. Set $\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x}_1 & \cdots & \tilde{x}_d \end{bmatrix}^\top := \eth_p(\mathbf{y})$
5. Choose a pseudorandom number $\alpha \in N(0,1)$;
6. Set $\tilde{x}_d := \alpha$
7. Set $R = e_{1,d}^\top$
8. **for** $j = 2,\ldots,d$ **do**
   Find $1 \leq k_1,\ldots,k_{n_j} \leq d$ such that $|\tilde{x}_j - \tilde{x}_{k_m}| \leq \varepsilon$, for each $1 \leq m \leq n_j$
   if $k_1 = j$ **then**
   Set $R_0 := (1/n_j)\sum_{l=1}^{n_j} \hat{e}_{l,d}^T$
   Set $R := \begin{bmatrix} R \\ R_0 \end{bmatrix}$
   end if
9. **end for**
10. Set $R_{p,L}(n) := R$

return $R_{p,L}(n)$

B. Structured coupling matrix identification algorithm

Given discrete-time dynamic financial model $(\Sigma,\mathscr{T})$ and a structured data sample $\Sigma_T \subset \Sigma$, we can apply Algorithm A.2 and Algorithm A.1, in order to to compute the otuput coupling matrix that can be used to obtain an approximate representation of the evoulution operator $\mathscr{T}$, corresponding to the orbit Σ. For this purpose one can use the following Algorithm.

V. NUMERICAL SIMULATIONS

In this section, we will present some numerical simulations computed using the **SPORT** toolset available in[?] , which was developed as part of this project. The toolset consists of a collection of Matlab programs for structured sparse identification and numerical simulation of discrete-time dynamical financial systems.

The numerical experiments documented in this section were performed with Matlab R2023a 64-bit. All the programs written for real-world data reading, synthetic data generation, and sparse model identification as part of this project are available at[?] .

The numerical simulations reported in this section were computed on a Linux Ubuntu Server 20.04 PC equipped with an Intel Xeon E3-1225 v5 (8M Cache, 3.30 GHz) processor and with 40GB RAM.

Algorithm B.1 ARC Model: ARC model identification

Data: $\Sigma_N = \{x_t\}_{t=1}^T \subset \mathbb{R}^n$, $\boldsymbol{\pi} : G \to \mathbb{O}(n)$
Result: OUTPUT COUPLING AND COMPRESSION MATRICES: $\hat{W},\tilde{W},R_{p,L}(n)$

1. Choose or estimate the lag value L using auto-correlation function based methods
2. Set a tensor order value p
3. Compute compression matrix $R_{p,L}(n)$ applying Algorithm A.2
4. Compute matrices:
$$\mathbf{H}_0 := \mathbf{H}_{L,G}^{(0,p)}(\Sigma_T)$$
$$\mathbf{H}_1 := \mathbf{H}_{L,G}^{(1)}(\Sigma_T)$$
5. Approximately solve:
$$\hat{W}\left(R_{p,L}(n)\mathbf{H}_0\right) = \mathbf{H}_1$$
for $\hat{W}$ applying Algorithm A.1

return $\hat{W},R_{p,L}(n)$

A. Sparse autoregressive reservoir computers for dynamical nonlinear financial system behavior identification

In this section, we consider a nonlinear dynamical financial system described by the model

$$\begin{aligned} \dot{x}_1 &= x_3 + (x_2 - s)x_1, \\ \dot{x}_2 &= 1 - cx_2 - x_1^2, \\ \dot{x}_2 &= -x_1 - ex_3, \\ x_1(0) &= x_0, x_2(0) = y_0, x_3(0) = z_0. \end{aligned} \tag{V.1}$$

As observed in[1], systems of the form (V.1) can exhibit, among others behavior types, chaotic and eventually approximately periodic dynamic behavior depending on the configuration of parameters and initial conditions considered for (V.1).

1. *Chaotic behavior identification*

For $s = 3, c = 0.1, e = 1$, let us consider the initial conditions $x_0 = 2, y_0 = 3, z_0 = 2$. For this configuration, one can obtain synthetic time series data $\Sigma_{14633} \subset \mathbb{R}^3$ obtained by applying a fourth-order adaptive numerical integration method to (V.1) for the configuration determined by the previous choice of parameters, followed by a spline interpolation to guarantee that the elements in Σ_{14633} are uniformly distributed with respect to the time interval [0,120].

The training orbit's data set corresponding to the first 61% of the data in Σ_{14633}, together with the remaining data used for model validation, are illustrated in Figure 1. The factorization for the output coupling matrix $W = \hat{W}R$ determined by Theorems III.6 and III.7 are illustrated in Figure 2.

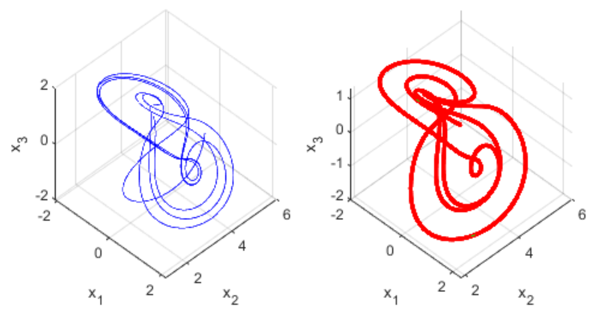


Figure 1: Training orbits data (left), validation orbits data (right). The green line corresponds to validation data, and the red dotted line corresponds to the model's predictions.

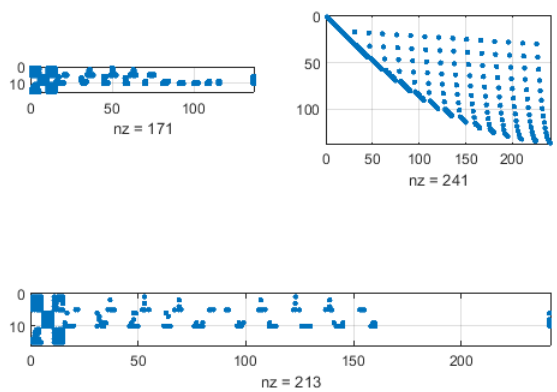


Figure 2: Matrix factors $\hat{W}$ (top-left) and R (top-right), output coupling matrix $W = \hat{W}R$ (bottom).

2. Eventually approximately periodic behavior identification

For $s = 0.5, c = 0.1, e = 0.1$, let us consider the initial conditions $x_0 = 1, y_0 = 1, z_0 = 1$. For this configuration, one can obtain synthetic time series data $\Sigma_{22901} \subset \mathbb{R}^3$ obtained by applying a fourth-order adaptive numerical integration method to (V.1) for the configuration determined by the previous choice of parameters, followed by a spline interpolation to guarantee that the elements in Σ_{22901} are uniformly distributed with respect to the time interval [0,120].

The training orbit's data set corresponding to the first 10% of the data in Σ_{22901}, together with the remaining data used for model validation, are illustrated in Figure 3. The factorization for the output coupling matrix $W = \hat{W}R$ determined by Theorems III.6 and III.7 are illustrated in Figure 4.

The computational setting used for the experiments per-

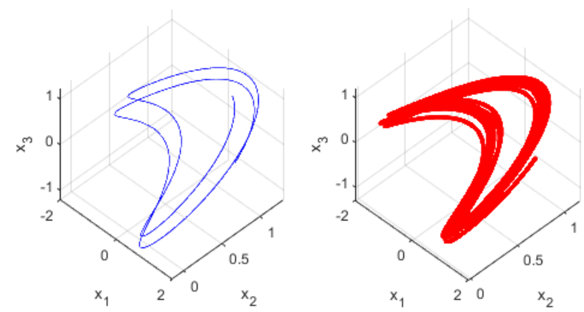


Figure 3: Training orbits data (left), validation orbits data (right). The green line corresponds to validation data, and the red dotted line corresponds to the model's predictions.

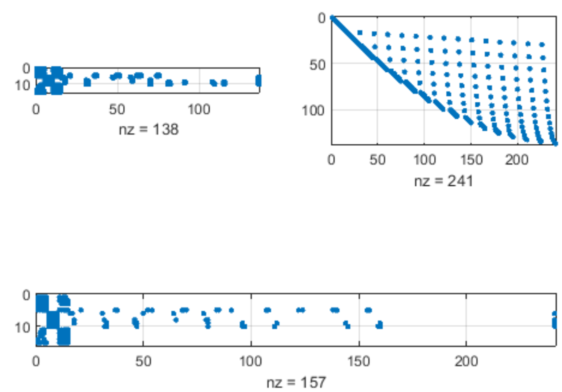


Figure 4: Matrix factors $\hat{W}$ (top-left) and R (top-right), output coupling matrix $W = \hat{W}R$ (bottom).

formed in this section is documented in the Matlab programs `SpaRC.m` and `NLFinancialSystem.m` in[?] that can be used to replicate these results.

3. Sparse identification of interest rate forecasting models

For this example, we will use Honduran lending interest rates time series dataset $\Sigma_{40} \subset \mathbb{R}$ recorded by the International Monetary Fund and the World Bank, that was collected between 1982 and 2021 and is available in[3] and also as part of the SPORT in[?]. For this experiment, we will consider two sparse autoregressive reservoir computer models. One of the models consists of a fully linear sparse autoregressive reservoir computer model determined by the expression

$$\left(1-\sum_{j=1}^{12} L^j\right)\lambda_t = e_\lambda(t),$$

where λ_t denotes the lending interest rates corresponding to year t, and where L denotes the lag operator determined for this case by the local condition $L^j\pi_t = \pi_{t-j}$, and where e_λ denotes the residual term determined by the covariance matrix and innovations corresponding to the model. The other model under consideration is a sparse nonlinear autoregressive reservoir computer model. Both models will be trained with the first 75% of the data in the time series sample Σ_{40}.

The reference data and the model's predictions are shown in figure 5.

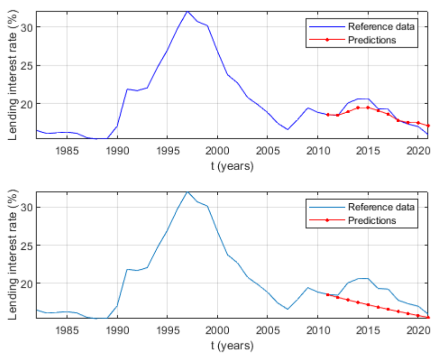


Figure 5: Reference signal (blue line) and model prediction (red dotted line) computed with sparse fully linear autoregressive reservoir computer model (top). Reference signal (blue line) and model prediction (red dotted line) computed with sparse nonlinear autoregressive reservoir computer (bottom).

The combination of fully linear with nonlinear sparse ARC models considered in this experiment is aimed to provide an example of the potential that this model combination has to capture both the short-term and long-term behavior of a given financial system under consideration.

The computational setting used for the experiments performed in this section is documented in the Matlab program `IntRatePredictor.m` in[?] , and this program can be used to replicate these experiments.

VI. CONCLUSION

The results in §III A and §III B in the form of algorithms like the ones described in §IV, can be effectively used for the sparse structured identification of financial dynamical models that can be used to compute data-driven predictive numerical simulations.

VII. FUTURE DIRECTIONS

Further implementations of the structured sparse model identification algorithms presented in this document to compute dynamic general equilibrium models will be the subject of future communications. The applications of sparse ARC models to inflation dynamics forecasting will be studied in future communications.

DATA AVAILABILITY

The programs and data that support the findings of this study will be openly available in the DyNet-CNBS repository, reference number[8], in due time.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

ACKNOWLEDGMENT

The structure preserving matrix computations needed to implement the algorithms in §IV, were performed with MATLAB 9.14.0.2206163 (R2023a) 64-bit, with the support and computational resources of the National Commission of Banks and Insurance Companies (**CNBS**) of Honduras.

REFERENCES

[1] Rajat Budhiraja, Manish Kumar, Mrinal K. Das, Anil Singh Bafila, and Sanjeev Singh. A reservoir computing approach for forecasting and regenerating both dynamical and time-delay controlled financial system behavior. *PLOS ONE*, 16(2):1–24, 02 2021.

[2] Matthew Dixon, Diego Klabjan, and Jin Hoon Bang. Classification-based financial markets prediction using deep neural networks, 2017.

[3] International Monetary Fund and Wold Bank Data Sets. Lending interest rates data for Honduras: 1982 - 2021. `https://data.worldbank.org/indicator/FR.INR.LEND?end=2021&locations=HN&start=1982`. License: CC-BY 4.0.

[4] Daniel J. Gauthier, Erik Bollt, Aaron Griffith, and Wendson A. S. Barbosa. Next generation reservoir computing. *Nature Communications*, 12(1):5564, Sep 2021.

[5] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 3rd edition, 1996.

[6] Balasubramanya T. Nadiga. Reservoir computing as a tool for climate predictability studies. *Journal of Advances in Modeling Earth Systems*, 13(4):e2020MS002290, 2021. e2020MS002290 2020MS002290.

[7] Justin Sirignano and Rama Cont. Universal features of price formation in financial markets: perspectives from deep learning, 2018.

[8] Fredy Vides. Dynet-cnbs: Network dynamics identification and reservoir computing tools with financial applications. 2022. https://github.com/FredyVides/SPORT.

[9] Ye Yuan, Xiuchuan Tang, Wei Zhou, Wei Pan, Xiuting Li, Hai-Tao Zhang, Han Ding, and Jorge Goncalves. Data driven discovery of cyber physical systems. *Nature Communications*, 10(1):4894, Oct 2019.